Electronic Version
Stylesheet Version v1.1.1

Description

METHOD AND APPARATUS FOR TRANSLATING DATA PACKETS FROM ONE NETWORK PROTOCOL TO ANOTHER

BACKGROUND OF INVENTION

[0001] *Technical Field*

[0002] The present invention relates to network communications in general, and, in particular, to a method and apparatus for translating data packets. Still more particularly, the present invention relates to a method and apparatus for translating data packets from one network protocol to another.

[0003] *Description of the Related Art*

[0004] In a mixed protocol network environment, a significant amount of processing power of a network router is spent on translating data packet information from one protocol to another. Packet translation is a relatively time-consuming process. For a network storage environment, it

is important to perform packet translation in an efficient manner in order to maintain an overall high system performance.

[0005] Conventionally, there are hardware solutions and software solutions for facilitating the packet translation process. One hardware solution is to construct packet headers with a state machine, but the problem with such solution is that there is no flexibility if network protocols change or migrate. Another hardware solution is to construct packet headers with a general purpose processor (GPP), but the problem with such solution is that long sequences of GPP instructions are required to execute protocol translations.

[0006] As for software solutions, information from the header or context block of a packet can be used to construct headers or frames. But the problem with such solution is that there are performance constraints from recalculations needed to be made for each translation.

[0007] Consequently, it is desirable to provide an improved method and apparatus for translating data packets between different network protocols.

Summary of Invention

[0008] In accordance with a preferred embodiment of the present invention, a set of translation templates is constructed.

The translation templates are then loaded into a translation template cache. In response to a data packet from a first network arriving at a translation router, an appropriate translation template is selected from the set of translation templates within the translation template cache according to the translation context of the data packet. Next, a new header for transmission into a second network is constructed by reading header fields of the data packet from the first network along with the appropriate translation template in the translation template cache. The data payload of the data packet from the first network is subsequently removed from the header of the data packet and then appended to the constructed header of the second network. Finally, the newly constructed data packet is transmitted to the second network.

[0009] All objects, features, and advantages of the present invention will become apparent in the following detailed written description.

Brief Description of Drawings

[0010] BRIEF DESCRIPTION OF THE DRAWINGS The invention itself, as well as a preferred mode of use, further objects, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative

embodiment when read in conjunction with the accompanying drawings, wherein: Figure 1 is a block diagram of a network environment to which a preferred embodiment of the present invention is applicable; Figure 2 is a block diagram of an apparatus for translating data packets between different network protocols, in accordance with a preferred embodiment of the present invention; Figures 3a-3b are block diagrams of a translation template cache, in accordance with a preferred embodiment of the present invention; Figure 4 is a block diagram of a translation template for facilitating the translation of a received data packet to a Fibre Channel protocol, in accordance with a preferred embodiment of the present invention; and Figure 5 is a high-level logic flow diagram of a method for translating data packets from one protocol to another by using the apparatus from Figure 2, in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION

[0011] DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Referring now to the drawings and in particular to Figure 1, there is depicted a block diagram of a network environment to which a preferred embodiment of the present invention is applicable. As shown, a network environment

10 includes a network 11 and a network 12. Each of network 11 and network 12 may have its own separate network protocol. For example, network 11 can be Ethernet while network 12 can be Fibre Channel, or network 11 can be Fibre Channel while network 12 can be InfiniBand. All communications between network 11 and network 12 are handled by a translation router 15. Translation router 15 is preferably a multi-protocol router. As a multi-protocol router, each port of translation router 15 is associated with a specific network protocol. For example, if translation router 15 has three ports, port 1 of translation router 15 can be associated with Fiber Channel, port 2 of translation router 15 can be associated with Transmission Control Protocol/Internet Protocol (TCP/IP), and port 3 of translation router 15 can be associated with an InfiniBand. Thus, translation router 15 is aware of the type of network protocol to which a network packet belongs based on the incoming port number from which the network packet comes. Similarly, translation router 15 is aware of the type of network protocol to which a network packet should be translated based on the outgoing port number to which the network packet is to be transmitted.

[0012] Within a network environment, such as network environ-

ment 10, the processing overhead can be relatively high because of the large number of data packets that needed to be dealt with. However, within the network environment, there are also a couple of features that can be used to alleviate the above-mentioned problem. First, although there are numerous network protocols that are being in-use, the total number of different network protocols are relatively small and the specifics of each network protocol are well defined by various governing standards. Second, most network protocols use a header structure ahead of the payload in a packet containing packet description information such as network source and destination addresses, type of packet, etc. The present invention leverages the above-mentioned two attributes in order to pre-build the necessary set of translation template caches as well as constructing the algorithms necessary to fill a particular destination translation template caches from any of the possible input network protocols.

[0013] With reference now to Figure 2, there is illustrated a block diagram of an apparatus for translating data packets from one network protocol to another, in accordance with a preferred embodiment of the present invention. As shown, an apparatus 20 includes a translation template cache 21

and a translation engine 22. Translation engine 22 is coupled to a general purpose processor 25, and general purpose processor 25 is coupled to an instruction cache 23, a data cache 24 and a system memory 26. Apparatus 20 is preferably located within a translation router, such as translation router 15 from Figure 1. Translation template cache 21 contains translation templates for facilitating the translation of data packets from one network protocol to another. Basically, each translation template within translation template cache 21 represents a unique network protocol supported by the translation router. The use of translation templates requires far lesser instructions to perform a network protocol translation as compared to using general purpose processor 25.

[0014] Translation engine 22 may be implemented as software, firmware, hardware, or combinations thereof, and is capable of executing operations necessary to manipulate information in packet headers for protocol translations. Translation template cache 21 may be implemented in a dedicated onmemory, whether as one or more locked cache lines or a private memory accessible to translation engine 22.

[0015] Referring now to Figure 3a, there is illustrated a block di-

agram of translation template cache 21, in accordance with a preferred embodiment of the present invention. As shown, translation template cache 21 contains multiple entries, each entry contains a translation template associated with a network protocol supported by a translation router, such as translation router 15 from Figure 1. Each translation template is preferably 64 bytes wide, which should be long enough to contain headers of data packets of various network protocols.

[0016] A translation template is required for each protocol translation. For example, consider a network A and a network B having a protocol A and a protocol B, respectively. One translation template is required for generating data packets suitable for transmission in network B from data packets received from network A. Another translation template is required to generate data packets suitable for transmission in network A from packets received from network B. If there is a third network C having a network protocol C needed to be supported, four more translation templates must be constructed and loaded into the translation template cache. Basically, two translation templates are required for protocol translations between network A and network C, and two translation templates are required for

translations between network B and network C.

[0017] Five different translation templates associated with their respective network protocols are shown in Figure 3b. As shown, entry 1 is a translation template for Fibre Channel frame type #1, entry 2 is a translation template for Fibre Channel frame type #2, entry 3 is a translation template for InfiniBand local routing, entry 4 is a translation template for InfiniBand global routing, and entry 5 is a translation template for TCP/IP.

[0018] The details of a translation template can be illustrated with an example. The control and information bits of a typical Fibre Channel frame header include:D_ID:destination identification (ID)S_ID:source IDTYPE:protocol of frame content (*e.g.*, 0000 1000 = SCSI FCP)F_CTL:flow controlSEQ_ID:sequence IDDF_CTL:data_field controlSEQ_CNT:sequence countOX_ID:originator exchange IDR:responder exchange IDParameter:carries information specific to Link_Control framesThus, a translation template for facilitating the translation of a received data packet to a Fibre Channel protocol should include most, if not all, of the above-mentioned control and information bits. With reference now Figure 4, there is illustrated a block diagram of a

translation template for facilitating the translation of a received data packet to a Fibre Channel protocol, in accordance with a preferred embodiment of the present invention. As shown, a translate template 40 includes a destination ID field 41, a source ID field 42, a type field 43, a flow control field 44, a sequence ID field 45, a data control field 46, a sequence count field 47, an originator exchange ID field 48, and a responder exchanged ID field 49. As shown, the fields within translation template 40 correspond to the fields within the Fibre Channel frame header.

[0019] Referring now to Figure 5, there is depicted a high-level logic flow diagram of a method for translating data packets from one protocol to another by using apparatus 20 from Figure 2, in accordance with a preferred embodiment of the present invention. Starting at block 30, a group of translation templates are constructed and the translation templates are loaded into a translation template cache, as shown in block 31. The translation templates, such as translation template 40 from Figure 4, are preferably constructed during initial configuration of a system, and the translation templates are then loaded into the translation template cache. Otherwise, the translation templates can

be constructed "on-the-fly" as they are being loaded into the translation template cache.

[0020] Next, exchange attributes are identified by a translation router, such as translation router 15 from Figure 1, as depicted in block 32. At this time, network protocols are also identified, usually as part of the translation engine configuration, and the translation templates are verified as available for use. As mentioned previously, a multi-protocol translation router is capable of identifying the type of network protocol to which a network packet belongs based on the incoming port number the network packet comes from, and is also capable of identifying the type of network protocol to which a network packet should be translated based on the outgoing port number the network packet is to be transmitted.

[0021] As each data packet arriving into the translation router, an appropriate translation template is selected from the translation template cache according to the translation context of the data packet by an translation engine, as shown in block 33.

[0022] Appropriate header fields from the data packets from a first network are read and used by the translation engine along with the appropriate translation template in the

translation template cache to generate new headers for transmission into a second network, as depicted in block 34.

[0023] The data payload of the data packet from the first network is removed from the header of the data packet, and is then appended to the recently constructed header for the second network, as shown in block 35. The resultant data packet intended for the second network is then placed into a local memory of the network router.

[0024] The data packet intended for the second network is subsequently pulled out of the local memory within the network router. This is usually performed by using a direct memory access (DMA) or by the general-purpose processor, as depicted in block 36. The data packet intended for the second network is sent to the second network.

[0025] As has been described, the present invention proan improved method and apparatus for translating data packets between different network protocols. Control blocks are constructed in a system memory to facilitate translation from one protocol domain to another. They must be repeatedly accessed as part of the translation process. With the present invention, the control blocks are stored onin a manner that facilitates repeated use over a number of

data packets. Further, the onstorage is not bound to one protocol format or another. In this way, packets may be translated across such protocols as Fibre Channel, Ethernet and InfiniBand.

[0026] It is also important to note that although the present invention has been described in the context of a fully functional translation router, those skilled in the art will appreciate that the mechanisms of the present invention are capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media utilized to actually carry out the distribution. Examples of signal bearing media include, without limitation, recordable type media such as floppy disks or CD ROMs and transmission type media such as analog or digital communilinks.

[0027] While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changin form and detail may be made therein without departing from the spirit and scope of the invention.